

Biweekly Report 3

Improving the dynamic comparison framework and theorising digital organisms

Raunak Narwal

Department of Mathematical Sciences

Indian Institute of Science Education and Research, Mohali, 130406,
Punjab

October 24, 2025

General Modular Framework and how to run the files on local machine

We integrated the two files, computing topological similarity and dynamic comparison, into a single automated workflow, *runall.py*. This script takes two network files (in *.kgml* or *.csv*) as input and runs both persistence and dynamic analyses. For all the uses one can type python *runall.py* to see the help.

```
(venv) C:\Users\rishu narwal\Desktop\MPI-CBG>python run_all.py
usage: run_all.py [-h] --graphA GRAPHA --graphB GRAPHB [--nameA NAMEA] [--nameB NAMEB] [--outdir OUTDIR] [--maxdim MAXDIM] [--p P]
                  [--convert-kgml]
run_all.py: error: the following arguments are required: --graphA, --graphB
```

To run the files on your local machine, clone this repository here, then run the following commands on your command prompt.

```
.venv\scripts\activate
pip install -r requirements.txt
python run_all.py --graphA --graphB --outdir
```

Improving the Dynamic Comparison framework

Our current dynamic comparison framework provides a simplistic functional baseline for simulating and comparing the behavior of two networks, it makes several simplifying assumptions that could limit its biological realism, current limitations include simplified reaction kinetics, all our reactions are currently happening with a rate constant of 1, assuming uniform first order kinetics, this oversimplifies the system. Incomplete Stoichiometry representation, each reaction decreases substrate concentration and increase

the product concentration by one unit, ignoring actual coefficients and multi substrate interactions. Another drawback of current workflow is that the framework aligns species solely by index order rather than by name or KEGG ID, which can produce misleading comparisons.

Some of the planned improvements include incorporating proper mass action and kinetics by parsing kinetic constants and rate laws from KGML or any external parameter file. Construct a full stoichiometric matrix that supports reactions of multiple substrates and products and that matches and compares only common species between networks based on KEGG IDs or names to ensure valid comparisons.

Pipeline Changes

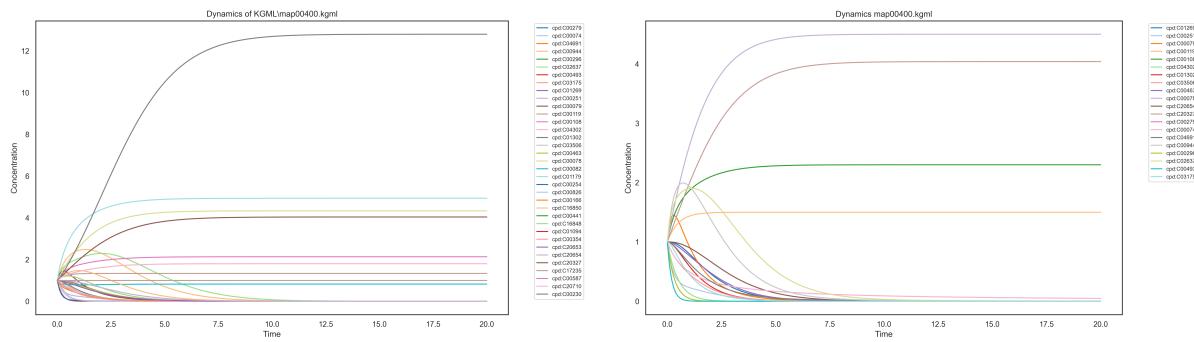
The refactored script now has a clear and modular design to improve debugging and future expansions. Stoichiometry is now fully supported by the KGML stoichiometry attribute, encoded into a proper matrix and reaction rates now follow mass action kinetics, replacing the previous assumptions. Parameter handling is made optional, monte carlo runs for robustness analysis. Species matching is improved by aligning either KEGG IDs or names of species, preventing misaligned trajectory comparisons. CLI is improved to contain following flags

```
--params FILE
--tmax Max simulation time
--npoints Number of time points
--method ODE solver (RK45, BDF)
--steady_threshold Stop early if system reaches steady state
--mc Monte Carlo runs for averaging
--atol, --rtol solver tolerances
```

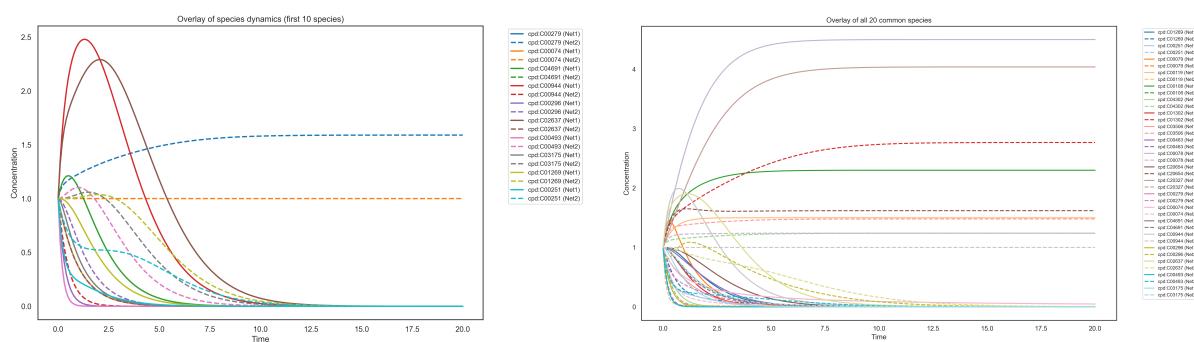
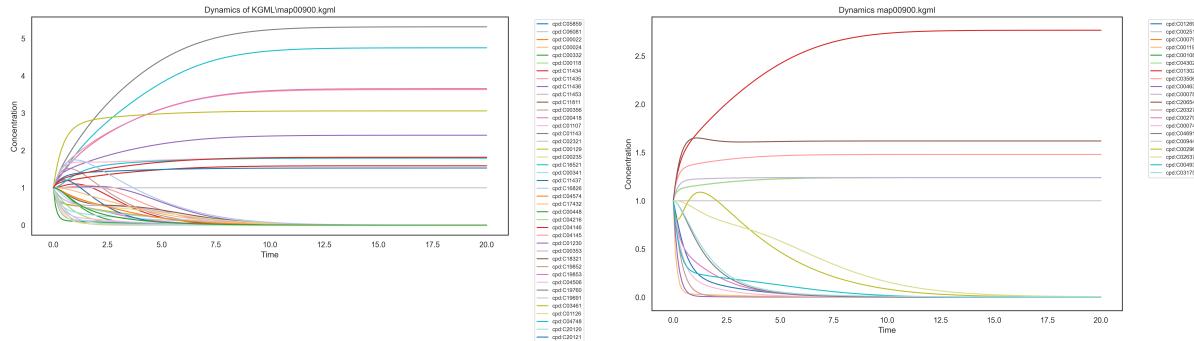
Results and Comparison with previous framework We compared map00400 and map00900 pathways with our updated dynamic comparison framework, the overall similarity metrics remained in the same range as previous, which suggests that underlying stoichiometry and reaction connectivity dominate the network dynamics more strongly than the kinetic form. The new framework shows a **DTW average of 6.2, MSE average of 2.14 and correlation average of 0.13, compared to 5.07 , 0.33 and 0.19 in the earlier version**. This increase in the averages reflects a higher dynamic distance, likely due to changes in the numerical solving precision and improved stoichiometric parsing.

Some compounds (*cpd:C00279, cpd:C00251, cpd:C00108*) maintained consistent patterns, with correlations above 0.9, which indicates that both frameworks capture the same trajectory trends despite numeric differences. *cpd:C00279 (ATP), cpd:C00944, cpd:C00493, and cpd:C03175* maintained strong positive correlations and low DTW/MSE values, which indicates that these species evolve in a similar temporal pattern across both the maps. These "similar acting" compounds likely corresponds to shared metabolic intermediates that play central role across pathways. This consistency of their trajectories suggests that the two network preserve common flux dynamics through these metabolites. Negative correlated species (*cpd:C00119, cpd:C00108, cpd:C00251*) indicate opposite flux trends between the networks, which tells pathway specific consumption or accumulation patterns. This often emerges because when some same metabolite serves distinct functions in separate systems, as a substrate in one map but a product in another.

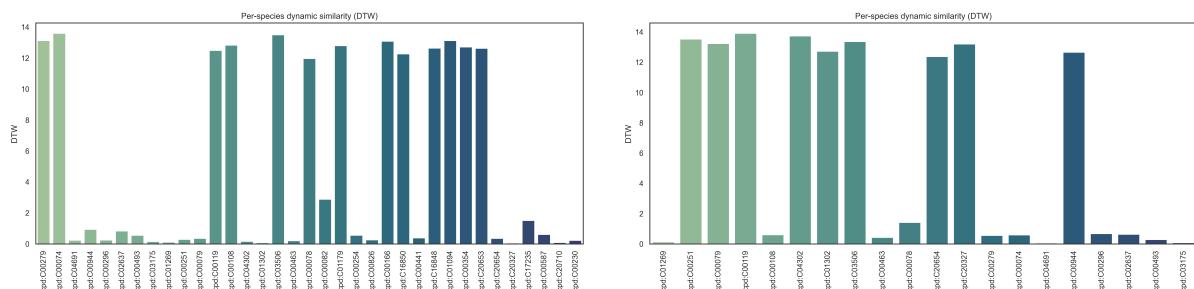
Comparing graphs and charts

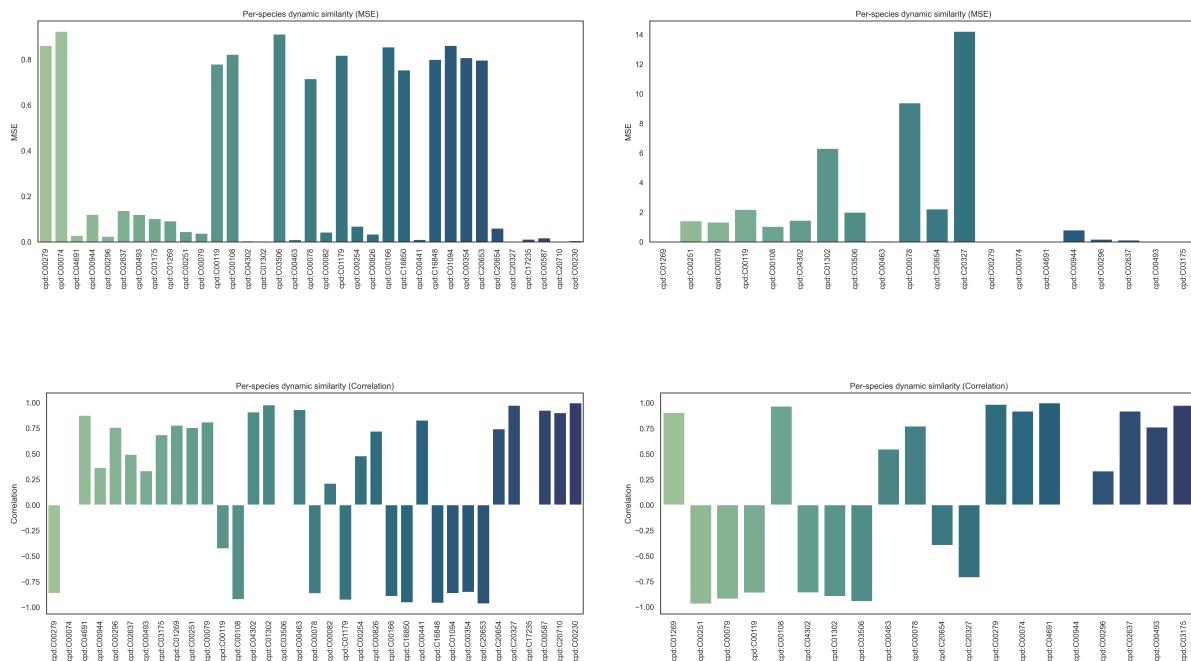


The improved framework only plots the common trajectories, this trajectory chart shows how concentrations of different compounds change over time. This is not comparing the two maps, it is the comparison of *map00400* trajectories plotted using two different dynamic frameworks. Same for *map00900* is given below.



The Overlay of Trajectories seems very different, because the previous framework shows overlay of first 10 species index views, while new framework plots all the common species calculated by taking intersection of the KEGGIDs and then sorting the list out.





DTW measures the similarity between two temporal sequences which may vary in speed, MSE quantifies the average squared difference between corresponding points in two trajectories, while correlation assesses the strength and direction of a linear relationship between them. All three metrics together provide a comprehensive view of how similarly the species behave over time in the two networks. A low DTW score indicates that the two species follow similar shapes even if one is behind the other in reaching equilibrium. A lower RMSE means that the two species are comparable in concentrations over time and a correlation near 1 indicates both species rise and fall together suggesting similar regulation or flux direction.

Michaelis Menten Kinetics

Michaelis-Menten kinetics is a model describing how the rate of an enzyme-catalyzed reaction depends on the substrate concentration. It assumes a reversible reaction where an enzyme and a substrate form a temporary complex, which then turns into a product. The Michaelis-Menten kinetics is not included in the current implementation because of several reasons. It does not yet produce a measurable difference from the mass action model because reactions are using default $k = 1$ as rate constants are not provided with the .KGML files. Only a few XML files contain rate constants, seeing this we can think of some ways to calculate rate constants or take them up from some other website. The saturation constant is given as $K_m = 1.0$. Without experimentally determined or literature-derived parameters, the Michaelis term

$$v = \frac{V_{max}[S]}{K_m + [S]}$$

reduces to near-linear behavior when $[S] \approx K_m$, mimicking the mass-action rate law. In real biochemical systems, V_{max} and K_m are enzyme-specific constants that define catalytic capacity. It is typically obtained from enzyme kinetics experiments, database values (for

example BRENDa, SABIO-RK) or parameter estimation via model fitting against time series metabolomics data (measurement and analysis of all small-molecule metabolites produced by metabolism within biological samples). Thus $k = 1$ is a placeholder, it acts like a neutral scaling factor to maintain numeric stability, it preserves network topology, it allows for dynamic comparison without requiring unknown biochemical data. The Monte Carlo runs are also producing the same results because initial conditions and parameters are all the same. MC becomes functionally meaningful when perturbations like random scaling and kinetic constants are introduced to evaluate robustness and sensitivity of dynamic similarity metrics.

Estimating rate constants

We can try to make our framework more biologically relevant by estimating rate constants using some approaches such as using computational parameter estimation methods, this might be time taking and maybe the trade off isn't fruitful enough, so we will take estimating rate constant as an agenda point for future reports. One common approach is to fit the reaction model to available reaction data for metabolite concentrations over time by minimizing the error between the simulations and observed data points, using nonlinear least squares or global optimization. If we do not have direct experimental data for a map, then ballpark estimates may be taken from related literature.

Introducing animations for better visualisation

We have tried various ways to see similarities between the two maps, but static plots can only convey so much information. To enhance interpretability, we have added animations that dynamically illustrate how species concentrations evolve over time in both networks. We saw that most of the action is happening in the first few seconds of the reactions, and later it reaches equilibrium. The links of both the videos are attached here.

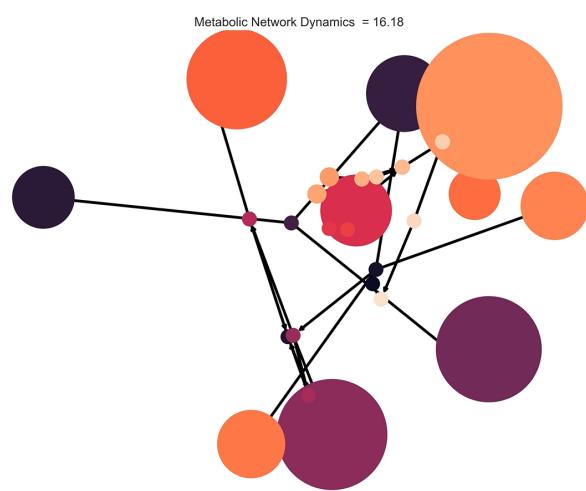
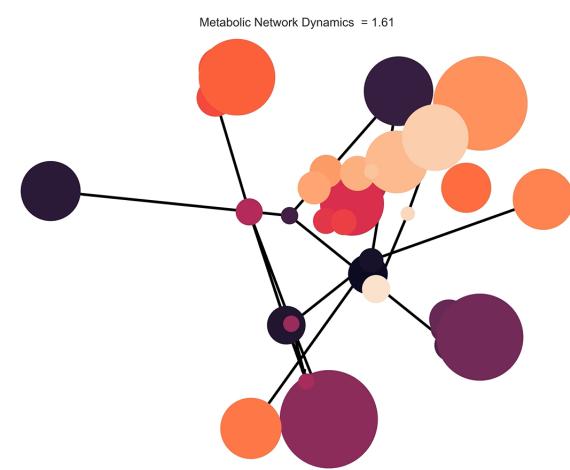
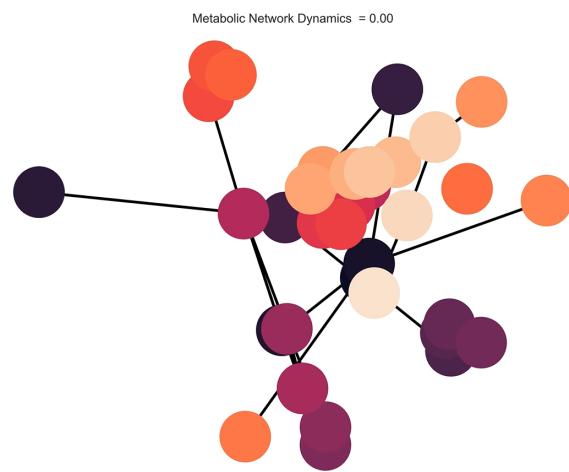
[Network1 Animation](#)

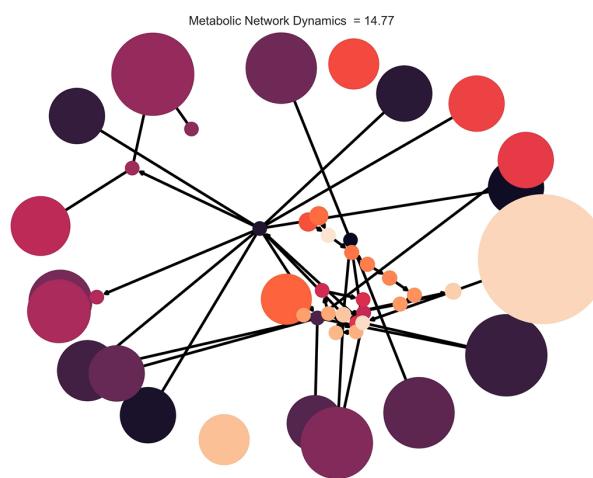
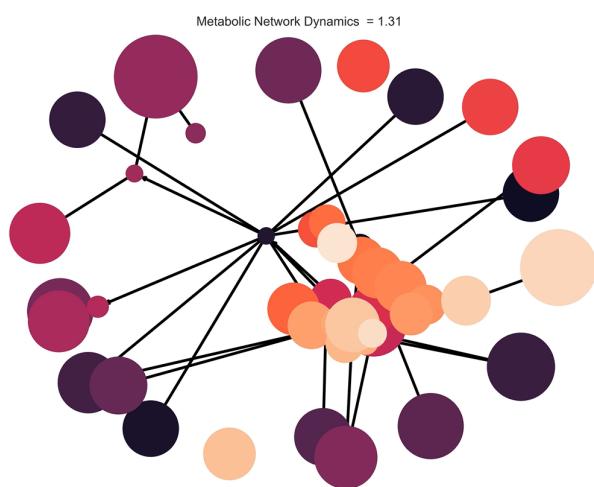
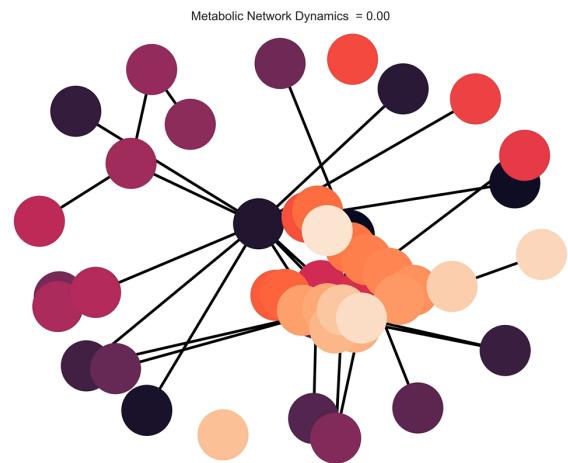
[Network2 Animation](#)

Something about animations makes things very much interesting and increase the understandability of the dynamics, we can see how different species are changing over time and how they are interacting with each other. This dynamic visualization helps to grasp complex temporal patterns that static plots and raw numbers might miss.

Label: Animations were made using matplotlib and seaborn libraries in python, the code is included in the repository. Nodes depict individual species and increase or decrease in size based on their concentration at each time point. Edges represent reactions, with thickness indicating flux magnitude.

Network1 Map00400



Network2 Map00400

Theorising digital organisms

I was thinking about how we can model digital organisms using networks, similar to biological systems, my planned digital organism, DIOR (digital organism) was a fun kind of project, mostly because i came across very complex research papers and i dropped the idea of making reality like digital organisms on my 12 GB RAM laptop. I say so because it took a whole year for a research team to move a single worm inside the screen. So my idea was to create DIOR using simple CRNs and keep text files as information and energy resources, we can have a virtual sun and a 2D grid for DIOR to live on. Let us dive deep on this Idea.

Here is a link to scientific paper titled "Simulation of the emergence of cell-like morphologies with evolutionary potential based on virtual molecular interactions"

Abstract: This study explored the emergence of life using a simulation model approach. The multiset chemical lattice model allows the placement of virtual molecules of multiple types in each lattice cell in a two-dimensional space. This model was capable of describing a wide variety of states and interactions, such as the diffusion, chemical reaction, and polymerization of virtual molecules, in a limited number of lattice cell spaces. Moreover, this model was capable of describing a wide variety of states and interactions, even in the limited lattice cell space of 100×100 cells. In this study, I assumed 18 types of virtual molecules, i.e., 18 virtual numbers that do not correspond to real molecules with chemical reactions represented by transformation of the numbers that occur with a specified reaction rate probability. Furthermore, it considered the energy metabolism and energy resources in the environment, and was able to reproduce evolution, in which a certain cell-like shape that adapted to the environment survived under conditions of decreasing amounts of energy resources in the environment. This enabled the simulation of the emergence of cell-like shapes with the four minimum cellular requirements, i.e., boundary, metabolism, replication, and evolution, based solely on the interaction of virtual molecules.

Ishida (2024) used a 100 X 100 lattice of virtual molecules that diffuse, react and polymerise. His model showed that cell like shapes could spontaneously form with basic life features such as boundary, metabolism, replication and evolution. In his model, molecules of different types transformed stochastically simulating chemical reactions and it could contain a informational ploymer molecule through cell divisions. The model even included a form of energy metabolism, when energy resources in the environment were scarce, only certain shapes survived having a certain polymer and reproduced, which demonstrates an emergent evolutionary process. Attached snapshot from his paper, illustrates his multiset chemical model.

www.nature.com/scientificreports/

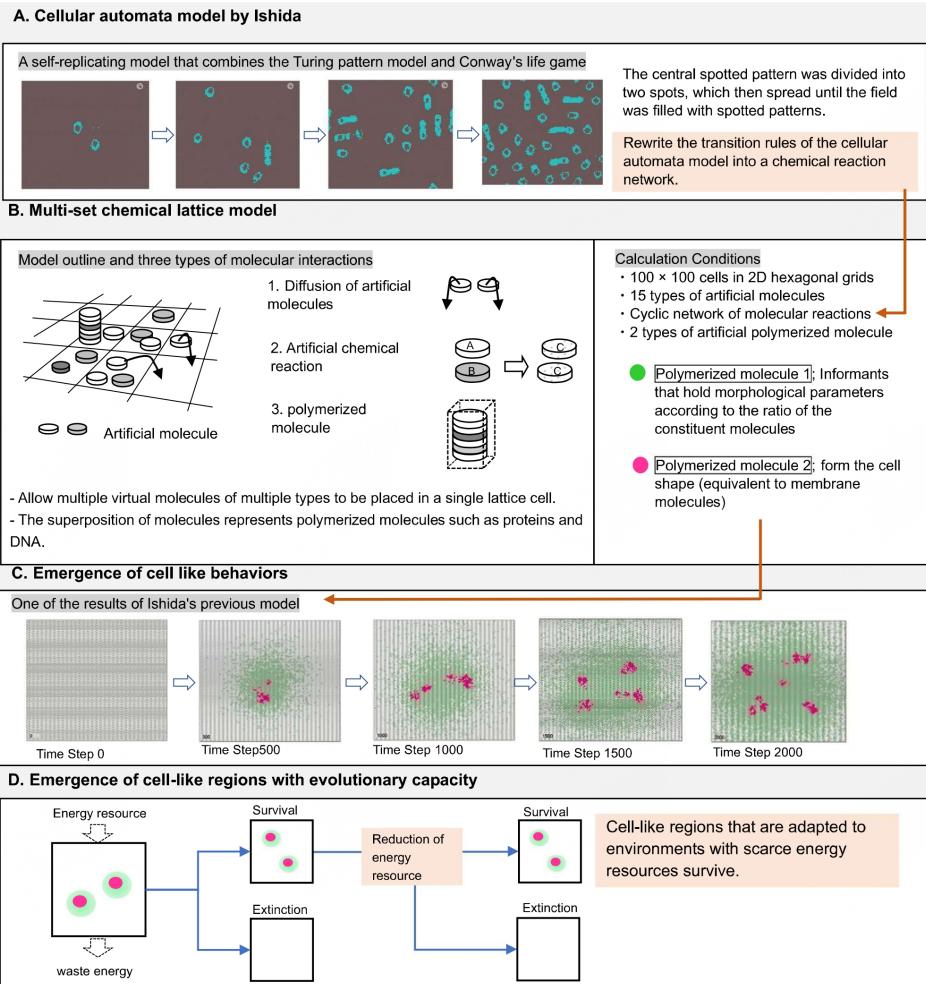
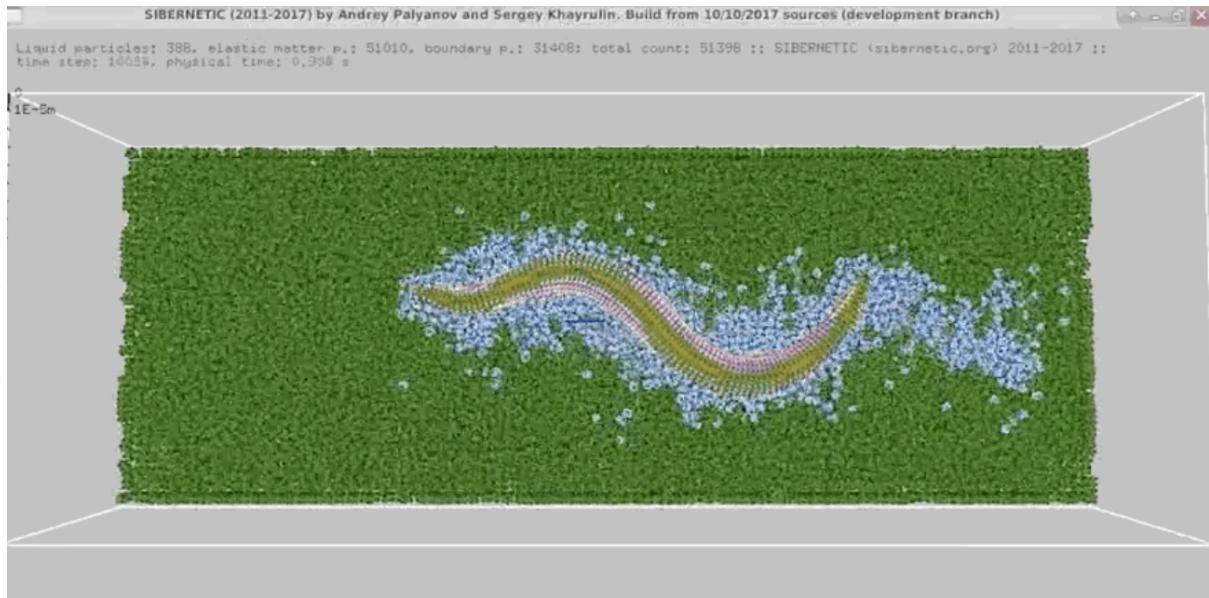


Figure 1. Schematic overview of the “Multiset chemical lattice model”. (A) Ishida²¹ improved the cellular automata model and found various shapes, such as self-replication. (B) This model allows the placement of multiple virtual molecules of multiple types in a single cell on a two-dimensional lattice space. The application of discrete stochastic transitions to each molecule allows the description of a wide variety of states and interactions in a limited number of cells, such as the migration (diffusion), transformation (chemical reaction), and linkage (polymerization) of virtual molecules. (C) Distribution of the two types of polymerized molecules. In the initial stage of the time step, an area of accumulation of red polymerized molecules was formed in the center of the lattice field, followed by the continuous disruption of this region of polymerized molecules. Concomitantly, it was confirmed that the information of the w value was retained in the green polymerized molecules and transmitted to the space. (D) The construction of an extended model that explicitly incorporates energy metabolism reproduced the phenomenon of evolution in which a certain cell-like shape that adapted to the environment survives under conditions of decreasing amounts of energy resources in the environment.

Development of such a system would take so much compute and time, the OpenWorm project (an attempt to simulate simple 1 mm worm) required many years of work and parallel team. In 2013 their team released only a rudimentary demo that showed just a few body parts moving and reported it would take another three to five months for a user friendly virtual worm was full runnable. Below is the 2019 docker container version for the worm.



So our system DIOR would be a very simplistic organism taking decisions based on certain chemical reaction networks, the system would need a 2D world grid with energy resources for example virtual sun that periodically adds energy and a way to let DIOR eat resources (.txt files in our case). Many digital life platforms treat CPU time as energy, for example in Avida system each organism gets CPU cycle to runs its individual code and performing tasks yield extra cycle to run, analogous to metabolizing food. We can simply credit DIOR with energy points when it ingests a text file. Another feature can be penalizing damage, for example if it runs out of energy or collides then it dies or loses health points. We can include asexual reproduction once DIOR reaches certain energy points. Our model needs to be simple and complex at the same time, for example DIOR eats .txt files then it itself gets stored as a .txt file (or JSON/.cvs as there are more code friendly) so that it can be eaten up by its descendants when no other .txt file is available. We need to have just the necessary variables, to simulate full isolation and behaviour of DIOR.

I am not too optimistic about this idea but we would try to do something experimental with the next biweekly report.

Discussion and Future Work

In this report we have improved our dynamic comparison framework and took a step towards something fun and exciting, an inside the screen organism. This idea is fun because it's make us seat on the god seat, the creator's seat. But we still have some pending work from previous weekly reports and future improvements on this report as well. Our working code now can handle any two kgml files and output persistence diagram,

graph maps, dynamic comparison metrics, overlay graphs and animated visualizations. However, there are still several avenues for further enhancement and validation:

- Implementing proper parameter estimation methods to calculate rate constants from literature or related databases to increase biological realism of dynamic simulations.
- Expanding the dynamic comparison framework to include stochastic simulation algorithms (e.g., Gillespie) to capture intrinsic noise in biochemical networks.
- Writing more theory and Developing DIOR (digital organism) prototype using simplified CRNs and energy/resource management within a 2D grid environment.
- Agenda items of biweekly report 2 (comparing permutations of the KGML files using API of KEGG databases or easier version of this could be , we can save 100s of KGML files in a specific folder, than our script compares these 100 files with each other that would be 4950 comparisons, and return to us just those files that fit our criteria of similarity.) and others listed in Biweely Report 2.

Event	Timeline
Biweekly Report 1	21 August
Biweekly Report 1.1	25 August
Mid semester exams	1-2 September
Biweekly Report 2	3 September 7 September
Mid semester exams	3 September 10 September
Awaiting Feedback	September-October
Mid semester II exams	10-16 October
Biweekly Report 3	21 September 24 October
Biweekly Report 4	7 November